

Week 4 Lab

For an example of how to do the exercises you can refer to my week 4 GitHub repo which is at:

<https://github.com/oit-gaden/Web-Development-2019-Winter/tree/master/Examples/Week4/app1>

Don't forget to push your code to GitHub in a folder called week3 and your Docker image to Docker Hub.

Exercise 1 - Install Vue CLI

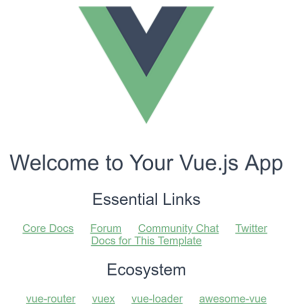
1. Install Vue CLI by running: **npm install -g vue-cli**
2. Run **vue --version** to make sure you have the Vue CLI installed. You may need to close and reopen a command/shell window first.

Exercise 2 - Create the sample Vue application

1. Create a folder named **week5** under your GitHub repo folder like you've done in past labs.
2. Enter the week5 folder and run: **vue init webpack webapp**.
 - a. You can choose the default answers when available.
 - b. Answer yes when asked if you want to install vue-router.
 - c. Answer no when asked if you want to use ESLint.
 - d. Answer yes when asked if you want to set up unit tests.
 - e. Answer "Jest".
 - f. Answer "no" for e2e tests.
 - g. Answer "use NPM".
 - h. NPM will now install all of the packages needed.
3. Follow the instructions displayed after NPM runs:
 - a. **cd webapp**
 - b. **npm run dev**

Webpack will now build the sample app and then display the URL to use for viewing the sample app in a browser. Try viewing the sample app to make sure everything worked as expected.

You should see the following:



4. Stop the **"npm run dev"** command (cntrl-c on Windows).

Exercise 3 - Start creating your first Vue application

1. Open the **week5/webapp** folder in VSCode and navigate to the folder **"src/components"**.

There you will find the **"HelloWorld.vue"** component.

2. Rename **HelloWorld.vue** to **Home.vue**.
3. Open **Home.vue** and find where the **template**, **script** and **style** sections are.
4. In the script section change the name of the component to **"Home"**.
5. Remove the line with the contents: **"msg: 'Welcome to Your Vue.js App'"**
6. Remove all of the CSS styles between **<style>** and **</style>**.
7. Replace all of the HTML between **<template>** and **</template>** with **<h1>Hello World</h1>**
8. In VSCode navigate to the folder: **"src/router"** and open the file **"index.js"**.
9. Change all occurrences of **"HelloWorld"** to **"Home"**.

10. Go back to the command line/shell in the week5/webapp folder and run: **"npm run dev"**.
11. Enter the URL in the browser and you should see the following displayed:



Hello World

12. The Vue logo is coming from the App.vue component which is the "root" component of your application. Vue determines the "root" component from looking at main.js.
13. Leave the command **"npm run dev"** running in your console/shell window as you continue the lab. You should see it rebuild the application as you make changes.

Exercise 4 - Convert our week 3 home page to Vue

1. Copy the contents of your week 3 **index.html** (home page) file between the body tags to your **Home.vue** component inside the template tags.
2. Remove the anchor (a) tags along with the associated href attribute and replace them with span tags. Your page navigation will no longer work. We'll fix that later.
3. Surround the content with a div element. Vue requires that the template content have only one root element.
4. If you still have the npm run dev command running and your browser open to the URL then you should see your Web app display change as you make content/style changes.
5. Remove the img from **src/App.vue**.
6. The style in **src/App.vue** will apply to your entire Web app since it is defined in the root component. You can remove or add any

styling there that you want to apply to your entire Web app.

7. Copy your image to the **src/assets** folder and fix the img reference in the **Home.vue** template.
8. Copy the CSS styles that applied to **index.html** content to the style section of **Home.vue**.
9. You should now see your "home page" in the browser displayed like you had it in week 3.

Exercise 5 - Convert your week 4 student page to Vue

1. Create **src/components/Students.vue**.
2. Add the template, script and style elements.
3. Copy the contents of your week 3 **students.html** file between the body tags to your **Students.vue** component inside the template tags.
4. Remove the anchor (a) tags along with the associated href attribute and replace them with span tags. Your page navigation will no longer work.
We'll fix that later.
5. Surround the content with a div element. Vue requires that the template content have only one root element.
6. Copy the script element contents from **Home.vue** and change name to **Students**.
7. Copy the CSS styles that applied to **students.html** content to the style section of **Students.vue**.
8. You'll view the changes after we fix the page links.

Exercise 6 - Convert your week 3 persons page to Vue

1. Follow the same steps you did in Exercise 5 to convert your students page.

Exercise 7 - Fix the navigation links

1. Since the navigation links apply to your entire Web app, first move the navigation links you had at the top of **index.html** to the **App.vue** component

just above the **router-view** element. You can now remove them from **Students.vue** and **Persons.vue**.

2. Define routes in **src/router/index.js** like this:

```
routes: [  
  {  
    path: '/home',  
    name: 'Home',  
    component: Home  
  },  
  {  
    path: '/students',  
    name: 'Students',  
    component: Students  
  },  
  {  
    path: '/persons',  
    name: 'Persons',  
    component: Persons  
  }  
]
```

3. Add imports at the top of **src/router/index.js** for the **Students** and **Persons** components.

4. Change your "links" at the top of **App.vue** to look like this:

```
<template>  
  <div id="app">  
    <span v-on:click='goToHome'>Home</span>  
    <span v-on:click='goToPersons'>Persons</span>  
    <span v-on:click='goToStudents'>Students</span>  
    <router-view/>  
  </div>  
</template>
```

5. Change your script section in **App.vue** to look like this:

```
<script>  
export default {  
  name: 'App',  
  methods: {  
    goToHome: function() {  
      this.$router.push({ path: 'home' })  
    },  
    goToPersons: function() {  
      this.$router.push({ path: 'persons' })  
    },  
    goToStudents: function() {  
      this.$router.push({ path: 'students' })  
    }  
  }  
}  
</script>
```

6. The navigation links at the top of your app should function as expected.

Exercise 8 - Dynamically generate the contents of the table in the students view.

1. Examine the **Products.vue** component in my GitHub repository and use it to determine how to dynamically populate the table in the **Students.vue** component.

Exercise 9 - Build a Docker image with your built app

1. Copy the Dockerfile and the .dockerignore files from your week 3 folder
2. You must build your app first by running the command: **npm run build**
This will create a folder named **"dist"**. This folder contains your "built" app.
3. Edit the Dockerfile and to the following:

Replace the line: **"COPY ./build /usr/share/nginx/app/"** with **"COPY ./dist /usr/share/nginx/app/"**
4. Build the Docker image.
5. Run the Docker image to test.
6. Push your Docker image to Docker Hub.

Extra Credit - Implement the login logic from week 3

1. 10 points: Examine the **Registration.vue** component in my GitHub repository week 3 folder.
Use it to determine how implement the login validation login from week 3 in your Vue app.
2. 10 points: Implement a navigation bar like the one in my GitHub repository week 3.